
Complexité

1. Donner la complexité dans le pire des cas des scripts suivants :

Script 1	Script 2	Script 3
<pre>for i in range(n): for j in range(n): x += 1</pre>	<pre>for i in range(n): for j in range(i): x += 1</pre>	<pre>for i in range(n): for j in range(n): for k in range(n): x += 1</pre>
Script 4	Script 5	
<pre>for i in range(n): for j in range(i): for k in range(j): x += 1</pre>	<pre>for i in range(n): if i%2 != 0 for j in range(n): x += 1 for j in range(n): y += 1</pre>	<pre>for i in range(5,n): for j in range(i-5,i+5): x += 1</pre>

2. On considère la fonction suivante :

```
def Fonction(n):
    for i in range(n):
        Operations en  $O(\log(n))$ 
    Operations en  $O(n)$ 
    Operations en  $O(n^2)$ 
```

— Quelle est la complexité de $Fonction(n)$?

3. Montrer que les assertions suivantes sont exactes :

- $n(n-1)$ est $O(n^2)$
- $\max(n^3, 10n^2)$ est en $O(n^3)$

4. Peut-on écrire :

- $2^{n+1} \in O(2^n)$?
- $2^{2n} \in O(2^n)$?
- $f(n) = O(g(n)) \Rightarrow g(n) = O(f(n))$?
- $f(n) + g(n) = O(\min(f(n), g(n)))$?

5. On considère la fonction suivante :

```
def FonctionR(n):
    if n = 0:
        return 2
    return FonctionR(n-1) + FonctionR(n-1)
```

- Quelle est la complexité de $FonctionR(n)$?
- Si on remplace l'instruction $return FonctionR(n-1) + FonctionR(n-1)$ par $return 2 * FonctionR(n-1)$, la complexité de la fonction $FonctionR$ est-elle modifiée ?

6. Soit les trois fonctions suivantes où n est un entier :

<pre>def Syr(n): while n > 1: n = n//2 return n</pre>	<pre>def Syrac(n): while n > 1: if n%2==0: n = n//2 else: n = 1 return n</pre>	<pre>def Syracuse(n): while n > 1: if n%2==0: n = n//2 else: n = 3*n +1 return n</pre>
--	---	---

- Donner les suites des valeurs successives prises par la variable n lors des appels des trois fonctions précédentes si $n = 32$ et $n = 20$.
 - Quelles sont les complexités (meilleur et pire des cas) de ces fonctions ?
7. Comparer la complexité des algorithmes de tri (sélection, à bulles, insertion, fusion et rapide).